

# SIEMENS

PATENT  
Attorney Docket No. 2002P11020WOUS

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Inventor:	Heller, Rainer	)	Group Art Unit:	2191
		)		
Serial No.:	10/527,914	)	Examiner:	Wu, Junchun
		)		
Filed:	03/16/2005	)	Confirmation No.:	9070

Title: SYSTEM FOR PREPARING A STANDARD FRAMEWORK FOR AUTOMATION  
DEVICES

Mail Stop Appeal Brief - Patent  
**Commissioner For Patents**  
P.O. Box 1450  
Alexandria, VA 22313-1450

### APPELLANTS' BRIEF UNDER 37 CFR 41.37

Sir:

This brief is in furtherance of the Notice of Appeal filed in this application on 27 October  
2009.

(Please proceed to the following page.)

1. REAL PARTY IN INTEREST - 37 CFR 41.37(c)(1)(i)

The real party in interest in this Appeal is the assignee of the present application, Siemens Aktiengesellschaft.

2. RELATED APPEALS AND INTERFERENCES - 37 CFR 41.37(c)(1)(ii)

There is no other appeal, interference or judicial proceeding that is related to or that will directly affect, or that will be directly affected by, or that will have a bearing on the Board's decision in this Appeal.

3. STATUS OF CLAIMS - 37 CFR 41.37(c)(1)(iii)

Claims cancelled: 1 – 20, 22, 23, 30 and 37.

Claims withdrawn but not canceled: None.

Claims pending: 21, 24 – 29, 31 – 36 and 38.

Claims allowed: none.

Claims rejected: 21, 24 – 29, 31 – 36 and 38.

The claims on appeal are 21, 24 – 29, 31 – 36 and 38. A copy of the claims on appeal is attached hereto in the Claims Appendix. Appellants respectfully appeal the final rejection of claims 21, 24 – 29, 31 – 36 and 38.

4. STATUS OF AMENDMENTS - 37 CFR 41.37(c)(1)(iv)

In response to the Final Office Action mailed 27 April 2009, Appellants filed a Response without claim amendment under 37 CFR 1.116 on 24 June 2009, requesting withdrawal of the final rejection. The Advisory Action mailed 22 September 2009 indicated that the Response did not change the status of the rejected claims and confirms disagreement between Appellants and the Examiner as to what the prior art discloses.

## 5. SUMMARY OF THE CLAIMED SUBJECT MATTER- 37 CFR 41.37(c)(1)(v)

With reference by page and line numbers to the detailed description, and with reference to the figures, the following summarizes one or more exemplary embodiments described in the Specification and which are covered by specific claims, but it is to be understood that the claims are not so limited in scope.

### 5A. CONCISE EXPLANATION OF SUBJECT MATTER DEFINED IN INDEPENDENT CLAIM 21

With reference generally to Figure 1 and Figure 2, **independent claim 21** is directed to a storage medium which stores a software system (2, 3<sub>i</sub>, 4, 5<sub>i</sub>, 9) for providing a programming environment to create device-independent functionality among automation devices 6<sub>i</sub> in an automation system of the type including a plurality, n, of automation devices 6<sub>i</sub> (page 2, lines 20 – 29; page 5, lines 10 – 23; page 8, lines 13 - 30). The system includes one or more automation engineering editors 12 for generating solutions for multiple ones of the automation devices (page 8, lines 32 – 36; page 9, lines 9 – 13 and 19 - 29). One or more compilers (5<sub>i</sub>, 11) translate the solutions into an intermediate language 10 in a runtime framework 4 (page 8, lines 19 – 22) for further translation (e.g., with system-specific compilers 5<sub>i</sub>) into different instructions for automation devices 6<sub>i</sub> in different automation systems (page 9, lines 19 – 29). An automation device-specific adapter 3<sub>i</sub> for each of the automation devices 6<sub>i</sub> provides a translation of a solution into instructions which can be interpreted by an automation device 6<sub>i</sub> in a different automation system (page 8, lines 17 – 21; page 10, lines 3 – 6). The software system provides encapsulation of specific functions of the automation devices and providing a base functionality of the automation devices (page 2, lines 20 – 29). The editors 12 and compilers 5<sub>i</sub> provide an automation functionality in a standard framework 4 for application among automation devices 6<sub>i</sub> having different command sets for being programmed (page 1, lines 17 – 21).

5B. CONCISE EXPLANATION OF SUBJECT MATTER DEFINED IN INDEPENDENT CLAIM 29

With further reference generally to Figure 1 and Figure 2, **independent claim 29** is directed to a method for providing device-independent functionality for automation devices 6<sub>i</sub> (page 2, lines 20 – 29; page 5, lines 10 – 23; page 8, lines 13 - 30). The method includes providing a compiler (5<sub>i</sub> , 11) for receiving solutions from one or more automation engineering editors 12 and translating the solutions into an intermediate language in a runtime framework 4 for further translation into different instructions for automation devices 6<sub>i</sub> in different automation systems (page 9, lines 19 – 29). An automation device-specific adapter 3<sub>i</sub> is provided for each of the automation devices. Each adapter 3<sub>i</sub> provides a translation of a solution from the intermediate language into instructions which can be interpreted by an automation device 6<sub>i</sub> in a different automation system (page 8, lines 17 – 21; page 10, lines 3 – 6). The software system provides encapsulation of specific functions of the automation devices (page 2, lines 20 – 29). The editors 12 and compiler 5<sub>i</sub> provide an automation functionality in a standard framework 4 for application among automation devices 6<sub>i</sub> having different command sets for being programmed (page 1, lines 17 – 21).

6. GROUNDS OF REJECTION TO BE REVIEWED UPON APPEAL - 37 CFR 41.37(c)(1)(vi)

Whether claims 21, 24 – 29, 31 – 36 and 38 are unpatentable under 35 U.S.C. Section 103 over Gloudeman et al. (U.S. 6,028,998) in view of Azarya et al. (U.S. 5,978,578).

7. ARGUMENT 37 CFR 41.37(c)(1)(vii)

7A. APPELLANTS TRAVERSE THE REJECTION OF EACH CLAIM 21, 24 – 29, 31 – 36 and 38 UNDER 35 U.S.C. SECTION 103.

With Regard to the Art Rejections, Patentability of Each Claim is to be Separately Considered

Appellants urge that patentability of each appealed claim should be separately considered. General argument, based on deficiencies in the rejection of independent claims 21 and 29 under Section 103 demonstrates patentability of all dependent claims. However, none of the rejected claims stand or fall together because each dependent claim further defines a unique combination that patentably distinguishes over the art of record. For this reason, to the extent the dependent claims are each argued separately, the Board is requested to consider each argument presented with regard to each dependent claim. Argument demonstrating patentability of each dependent claim is presented under subheadings identifying each claim by number.

7A(1) THE REJECTION OF INDEPENDENT CLAIM 21 UNDER SECTION 103 BASED ON GLOUDEMAN ET AL. IN VIEW OF AZARYA IS IN ERROR.

Application of the Gloudeman and Azarya references to reject claim 21 under Section 103 results in deficiencies that render the rejection incorrect. Claim 21 is directed to a storage medium which stores a software system for providing a programming environment to create device-independent functionality among automation devices in an automation system of the type including a plurality of automation devices. The rejection cites col. 2, lines 20-28 of Gloudeman for describing a framework which leaves the user free to create an application without having to “worry about device-dependent details.” However, providing a framework wherein the user does not have to worry about details **does not mean that the framework creates a device-independent functionality** among automation devices.

Claim 21 requires that the

“editors and compiler [provide] an automation functionality in a standard framework for application among automation devices having different command sets for being programmed.”

In this regard, the passage of Gloudeman relied upon for this claimed subject matter does not disclose or suggest what is claimed. Specifically, col. 2, lines 27-32 discloses a framework which is consistent for building an automation framework – “a scalable architecture that will function on a wide range of processor platforms ...” However, none of the citation discloses or suggests the feature of providing a functionality among automation devices having **different command sets**. This prior art clearly does not disclose such. Rather, as best understood, the citation from Gloudeman concerns consistent use of a single framework when the processor platforms may vary in size from a small controller to an operator work station. There is no basis to argue that the context in which the word “consistent” is used (in Gloudeman) contemplates an ability to apply a framework among devices having different command sets. The use of the word “standard” in claim 21 is in a different context than the use of the word “consistent” in the passage cited from the Gloudeman reference.

However, the final office action argues that Gloudeman implicitly discloses the above-quoted feature, further citing col. 4, lines 22-40. While the cited passage employs the word “commands” and discloses changing the commands (“flexibly changed for generating p-code”), this is not the same as disclosing automation devices having **different command sets** for being programmed. That is, the prior art passage only refers to changing the commands and does not refer to using different command sets for different devices. For this reason the references cannot be combined to re-create the invention from the prior art. The combination does not result in what is claimed and there is no suggestion or motivation for modifying the prior art to provide the claimed combination. There can be no basis for obviousness.

Furthermore, the Azarya reference has been relied upon because, as stated at page 4 of the office action, the Examiner acknowledges that Gloudeman does not disclose

“a compiler for translating the solutions into an intermediate language in a runtime framework for further translation into different instructions for automation devices in different automation systems.”

However, this deficiency cannot be remedied by Azarya (e.g., the citation at col. 3, lines 16 – 32) because the compiler of Azarya does not “translate solutions into an intermediate language in a runtime framework for further translation into different instructions for automation devices in different automation systems” as argued in the rejection. A reading of the cited passage (col. 3, lines 16-32 of Azarya; col. 4, lines 56-67) confirms that the reference merely discusses generating a p-code to be executed on a target system, i.e., one system. There is no suggestion of “translation into different instructions for automation devices in different automation systems.” Despite citation of many other lines of text from the Azarya reference, the prior art does not disclose what is claimed. In fact, the rejection seems to acknowledge this by asserting that the reference “implicitly discloses” the claimed feature.

With further regard to the above-quoted recitation, it is noted that the rejection claims that the passages at col. 4, lines 56-62 and/or lines 63-67 and/or col. 5, lines 30-37 somehow disclose the feature of “for further translation into different instructions for automation devices in different automation systems” but, despite many citations to the Azarya reference, this prior art does not at all disclose or imply this feature.

The standard for obviousness requires a prima facie case and this cannot be had with handwaving or unsupported allegations that the subject matter is **implicitly disclosed**.

Another example of a deficiency in the prior art combination relates to the requirement in claim 21 for

an automation device-specific adapter for each of the automation devices, each adapter providing a translation of a solution into instructions which can be interpreted by an automation device in a different automation system ...

*It is not at all understood how or why the Examiner concludes* that the citations to Azarya (col. 4, lines 56-67, col. 5, lines 30-37) can relate to the above recitation. Appellants have requested the Examiner provide express support for the rejection in the prior art or withdraw the rejection. There has been neither a provision of support nor a withdrawal of the rejection.

The prior art combination cannot result in the invention of claim 21 for at least two reasons: first the above noted deficiencies preclude a rejection; and second, notwithstanding the deficiencies, the combination is only a hindsight reconstruction according to the applicants' teachings. That is, none of the prior art identifies the problems the applicants solve, or any other reason to make the combination which is claimed. The features of the storage medium according

to independent claim 29 include a software system for providing a “programming environment to create device-independent functionality among automation devices in an automation system ...”

The references are each directed to different problems, and the piecemeal extraction of features from each amounts to no more than a hindsight reconstruction of the invention.

As noted multiple times during this examination, **MPEP Section 2142** describes the tendency to resort to hindsight based on the applicants’ disclosure and that this is often difficult to avoid due to the nature of the examination process. Nonetheless, such **hindsight must be avoided**. This hindsight is especially apparent based on the effort to read applicant’s compiler (see claim 21) on the disclosure of Azarya. It is only with the claimed arrangement (see claim 21) that one solution can be developed for multiple automation devices and then translated into instructions for automation devices in different automation systems.

Even if there was an ability to meet all of the terms of the claims by combining the references as proposed in the new rejection, there would still be no teaching to form a basis for obviousness. There is no precedent to use the components to achieve the claimed functions. No one, without knowledge of the present application, would look to these references to create that which is now claimed. For all of these reasons the rejection of claim 21 should be overturned.

#### 7A(2) THE REJECTION OF INDEPENDENT CLAIM 29 UNDER SECTION 103 BASED ON GLOUDEMAN ET AL. IN VIEW OF AZARYA IS ALSO IN ERROR.

It is noted that the rejection of claim 29 is bundled with the rejection of claim 21 while the rejection refers to recitations of claim 21. Application of the Gloudeman and Azarya references to reject claim 29 under Section 103 also results in deficiencies similar to or the same as argued above with respect to claim 21 such that the rejection is also incorrect.

Claim 29 is directed to a method for providing device-independent functionality for automation devices. It requires, among other features, that the

“editors and compiler [provide] an automation functionality in a standard framework for application among automation devices having different command sets for being programmed.”



In this regard, the passage of Gloudeman relied upon for this claimed subject matter does not disclose or suggest what is claimed. Specifically, col. 2, lines 27-32 discloses a framework which is consistent for building an automation framework – “a scalable architecture that will function on a wide range of processor platforms ...” However, none of the citation discloses or suggests the feature of providing a functionality among automation devices having **different command sets**. This prior art clearly does not disclose such. Rather, as best understood, the citation from Gloudeman concerns consistent use of a single framework when the processor platforms may vary in size from a small controller to an operator work station. There is no basis to argue that the context in which the word “consistent” is used (in Gloudeman) contemplates an ability to apply a framework among devices having different command sets. The use of the word “standard” in claim 21 is in a different context than the use of the word “consistent” in the passage cited from the Gloudeman reference.

However, the final office action argues that Gloudeman implicitly discloses the above-quoted feature, further citing col. 4, lines 22-40. While the cited passage employs the word “commands” and discloses changing the commands (“flexibly changed for generating p-code”), this is not the same as disclosing automation devices having **different command sets** for being programmed. That is, the prior art passage only refers to changing the commands and does not refer to using different command sets for different devices. For this reason the references cannot be combined to re-create the invention from the prior art. The combination does not result in what is claimed and there is no suggestion or motivation for modifying the prior art to provide the claimed combination. Citation of col. 3, lines 20-24 (see advisory action) does not provide any support in this regard. There can be no basis for obviousness.

Furthermore, the Azarya reference has been relied upon because, based on statements made with respect to claim 21 at page 4 of the office action, the Examiner appears to acknowledge that Gloudeman does not disclose

providing a compiler for receiving solutions from one or more automation engineering editors and translating the solutions into an intermediate language in a runtime framework for further translation into different instructions for automation devices in different automation systems

However, this deficiency cannot be remedied by Azarya (e.g., the citation at col. 3, lines 16 – 32) because, as already noted above with regard to claim 21, the compiler of Azarya does

not “translate solutions into an intermediate language in a runtime framework for further translation into different instructions for automation devices in different automation systems” as argued in the rejection. A reading of the cited passage (col. 3, lines 16-32 of Azarya; col. 4, lines 56-67) confirms that the reference merely discusses generating a p-code to be executed on a target system, i.e., one system. There is no suggestion of “translation into different instructions for automation devices in different automation systems.” Despite citation of many other lines of text from the Azarya reference, the prior art does not disclose what is claimed. In fact, the rejection seems to acknowledge this by asserting that the reference “implicitly discloses” the claimed feature.

With further regard to the above-quoted recitation, it is noted that the rejection claims that the passages at col. 4, lines 56-62 and/or lines 63-67 and/or col. 5, lines 30-37 somehow disclose the feature of “for further translation into different instructions for automation devices in different automation systems” but, despite many citations to the Azarya reference, this prior art does not at all disclose or imply this feature.

The standard for obviousness requires a prima facie case and this cannot be had with handwaving or unsupported allegations that the subject matter is **implicitly disclosed**.

Another example of a deficiency in the prior art combination relates to the requirement in claim 29 for

providing an automation device-specific adapter for each of the automation devices, each adapter providing a translation of a solution from the intermediate language into instructions which can be interpreted by an automation device in a different automation system

The citation to Azarya (col. 4, lines 56-67, col. 5, lines 30-37) does not at all can relate to the above recitation.

The prior art combination cannot result in the invention of claim 29 for at least two reasons: first the above noted deficiencies preclude a rejection; and second, notwithstanding the deficiencies, the combination is only a hindsight reconstruction according to the applicants' teachings. That is, none of the prior art identifies the problems the applicants solve, or any other reason to make the combination which is claimed. The features of the storage medium according to independent claim 29 include a software system for providing a “programming environment to create device-independent functionality among automation devices in an automation system ...”

The references are each directed to different problems, and the piecemeal extraction of features from each amounts to no more than a hindsight reconstruction of the invention.

In the Advisory Action mailed 22 September 2009, the response to argument after final rejection appears to do little more than restate the argument or citations in the final rejection. The above deficiencies were not addressed.

Even if there was an ability to meet all of the terms of claim 21 by combining the references as proposed in the new rejection, there would still be no teaching to form a basis for obviousness. There is no precedent to use the components to achieve the claimed functions. No one, without knowledge of the present application, would look to these references to create that which is now claimed. For all of these reasons the rejection of claim 29 should be overturned.

#### 7B. EACH OF THE DEPENDENT CLAIMS IS ALLOWABLE UNDER SECTION 103.

Each dependent claim is patentable as depending from the respective dependent claim argued above. In addition, further argument is provided for dependent claim 25 below.

#### 7B(1) THE REJECTION OF DEPENDENT CLAIM 25 UNDER SECTION 103 BASED ON GLOUDEMANN ET AL. IN VIEW OF AZARYA IS ALSO IN ERROR.

According to claim 25, the software system provides technological objects for automation devices and, when the system includes  $m$  editors and  $n$  automation devices, at most, only  $n + m$  compilers are required to implement the solution.

The compiler, providing an intermediate language, in combination with an "automation device-specific adapter for each of the automation devices" not only results in each adapter providing a translation of a solution into instructions, but also reduces the quantity of compilers used for developing an automation solution. Otherwise, each programming language of each editor used would have to be translated with a special compiler for the target platform. The prior art combination used to reject the claims would still require  $n*m$  compilers for  $m$  editors and  $n$  automation devices. As a feature of the claimed invention, only  $n + m$  compilers are required to implement a specific automation solution. This is not at all recognized by the art of record.

Notwithstanding a complete absence of these features in the prior art, the rejection – without any citation in support thereof, concludes that the combined teachings of the prior art

Serial No. 10/527,914  
Atty. Doc. No. 2002P11020WOUS

result in only  $n + m$  compilers. To the contrary, the prior art combination used to reject the claims would still require  $n*m$  compilers for  $m$  editors and  $n$  automation devices. The rejection must be overturned.

#### 7C. CONCLUSION

Argument has been presented to demonstrate that the rejections under Section 103 are deficient and that dependent claim 25 further distinguishes over the prior art. The Examiner has argued rejections when claimed features are absent from or inconsistent with the applied combinations of art. Accordingly, none of the rejections can be sustained. For all of the above argued reasons, all of the rejections should be withdrawn and the claims should be allowed.

#### 8. APPENDICES

An appendix containing a copy of the claims involved in this appeal is provided herewith. No evidence appendix or related proceedings appendix is provided because no such evidence or related proceeding is applicable to this appeal.

Respectfully submitted,

Dated: Dec. 16, 2009

By: Janet D. Hood  
Janet D. Hood  
Registration No. 61,142  
(407) 736-4234

Siemens Corporation  
Intellectual Property Department  
170 Wood Avenue South  
Iselin, New Jersey 08830

## 9. APPENDIX OF CLAIMS ON APPEAL

21. A storage medium which stores a software system for providing a programming environment to create device-independent functionality among automation devices in an automation system of the type including a plurality of automation devices, the system comprising:

one or more automation engineering editors for generating solutions for multiple ones of the automation devices;

a compiler for translating the solutions into an intermediate language in a runtime framework for further translation into different instructions for automation devices in different automation systems; and

an automation device-specific adapter for each of the automation devices, each adapter providing a translation of a solution into instructions which can be interpreted by an automation device in a different automation system, the software system providing encapsulation of specific functions of the automation devices and providing a base functionality of the automation devices, the editors and compiler providing an automation functionality in a standard framework for application among automation devices having different command sets for being programmed.

24. The storage medium according to claim 21, wherein the software system is provided for developing control software in the automation system.

25. The storage medium according to claim 21, wherein the software system provides technological objects for automation devices and, when the system includes  $m$  editors and  $n$  automation devices, at most, only  $n + m$  compilers are required to implement the solution.

26. A system including the storage medium according to claim 21, further comprising: a memory for storing automation solutions for recurring tasks.

27. The system according to claim 26, adapted for using the Internet and/or an intranet for transmitting data.

28. The storage medium according to claim 21, wherein an automation-specifically designed programming language is used for developing control software for the automation system.

29. A method for providing device-independent functionality for automation devices, the method comprising:

providing a compiler for receiving solutions from one or more automation engineering editors and translating the solutions into an intermediate language in a runtime framework for further translation into different instructions for automation devices in different automation systems; and

providing an automation device-specific adapter for each of the automation devices, each adapter providing a translation of a solution from the intermediate language into instructions which can be interpreted by an automation device in a different automation system, the software system providing encapsulation of specific functions of the automation devices, the editors and compiler providing an automation functionality in a standard framework for application among automation devices having different command sets for being programmed.

31. The method according to claim 29, wherein automation functionality is provided independent of the automation device.

32. The method according to claim 29, wherein a development system is used for developing control software.

33. The method according to claim 29, further comprising: providing technological objects for the automation devices.

34. The method according to claim 29, further comprising: storing automation solutions for recurring tasks.

35. The method according to claim 28, wherein the Internet or an intranet is used for transmitting data.

36. The method according to claim 29 including providing a programming language automation-specifically adapted for developing control software.

38. The method according to claim 36, wherein compilers are provided for mapping the intermediate language onto a target platform.

Serial No. 10/527,914  
Atty. Doc. No. 2002P11020WOUS

10. EVIDENCE APPENDIX - 37 CFR 41.37(c) (1) (ix)

None



Serial No. 10/527,914  
Atty. Doc. No. 2002P11020WOUS

11. RELATED PROCEEDINGS APPENDIX - 37 CFR 41.37(c) (1) (x)

None